**שידור מידע**

**בין 2 מחשבים**

**באמצעות מצלמה**

20/02/2022

A picture containing desk, indoor, computer, mouse

Description automatically generated

ליאור אלבז 315248070

תוכן עניינים

1. עמוד שער – עמ׳ 1
2. מבוא – עמ׳ 3
3. דרישות קדם – עמ׳ 4
4. הוראות התקנה – עמ׳ 4
5. הוראות שימוש – עמ׳ 5
6. ארכיטקטורת תוכנה – עמ׳ 6-9
7. תהליך שליחת קובץ – עמ׳ 10
8. מקרי בדיקה – עמ׳ 11
9. פערים במצב הקיים – עמ׳ 11

2. מבוא

התוכנה היא תוכנה להעברת קבצים בין מחשבים, באמצעות המצלמה.  
התוכנה מורכבת משני חלקים – transmitter ו receiver.

מחשב אחד שמקבל את המידע ומחשב אחד שמשדר את המידע.  
המחשבים צריכים למקם את המצלמה שלהם כך שכל מצלמה של האחד רואה את המסך של המחשב השני.

בעת הרצת התוכנה המחשבים ישדרו אחד לשני QR Codes, ה transmitter ישדר ברקודים שמכילים את מידע הקובץ, ועוד metadata בשביל שה receiver יוכל לקרוא.  
ה receiver ישדר QR Codes שמסמנים איזה מספר הודעה הוא קרא.

בעצם אנו יוצרים שכבה פיזית חדשה להעברת מידע, אלטרנטיבית לשכבת הרשת שאנו מכירים במחשבים.

הפרוטוקול דומה מאוד לפרוטוקול מספר 2 שנלמד בקורס מבוא לתקשורת מחשבים.

3. דרישות קדם

על 2 המחשבים להכיל מצלמה מחוברת ופעילה. וכמובן מסך להציג את ה QR Codes.

על 2 המערכות צריך להיות מותקן interpreter של python גרסה 3.6 ומעלה.  
וכן יש להתקין את הספריות המצוינות בקובץ requirements.txt

כמו כן, במחשבי מאק יש לתת גישה למצלמה לתהליך שיריץ את קובץ הפייתון (לדוגמה ה terminal או ה IDE). ניתן להריץ את הפקודה הבאה בכדי שהטרמינל יבקש גישה למצלמה:

$ ffmpeg -ss 0.5 -f avfoundation -i “0” -t capture.jpg

על מנת להתחיל את השידור יש לכוון את המצלמה של כל אחד מן המחשבים אל המסך של משנהו.

4. הוראות התקנה

בכדי להתקין python על המערכת מומלץ לעקוב אחרי המדריך:

[Python 3 Installation & Setup Guide – Real Python](https://realpython.com/installing-python/)

בכדי להתקין את הספריות הדרושות, רצוי להשתמש במנהל החבילות pip.  
ולהריץ את הפקודה

$ pip install -r requirements.txt

[How to install Python packages with pip and requirements.txt | note.nkmk.me](https://note.nkmk.me/en/python-pip-install-requirements/)

5. הוראות שימוש

לאחר התקנת python3, כל הספריות וכיוון המצלמות.

נריץ את ה transmitter.  
לשם כך יש להריץ את קובץ transmitter.py ולתת כארגומנט ראשון את הנתיב לקובץ שאותו רוצים לשדר.  
לדוגמה:

$ python3 transmitter.py ~/my\_file.txt

כעת נריץ על המחשב השני את ה receiver.  
בכדי להריץ את receiver יש פשוט להריץ את קובץ receiver.py  
לדוגמה:

$ python3 receiver.py

6. ארכיטקטורת תוכנה

המערכת מבוססת על הרצת 2 סקריפטים ב python, אחד על כל מחשב.

הסקריפטים עצמם בנויים בכך שיש מחלקות Receiver ו Transmitter שלכל אחת מהם פונקציה מרכזית אחת שהיא מחצינה.  
זאת בעיקר בשביל להקל על הקריאה של הקוד, ועל ה testability שלו.

המערכת עושה שימוש במספר ספריות:

* אנו משתמשים בספריית OpenCV על מנת לזהות את קודי ה QR מתוך התמונה במצלמה.  
  כמו כן בשביל להציג את הברקודים על המסך
* אנו משתמשים ב Numpy ובספריית qrcode בכדי לייצר את הברקודים

כמובן שאי אפשר לשדר את כל הקובץ באותו QR Code, מכיוון שיש מגבלה על כמות המידע ש QR Code יכול להציג, וכמות המידע שהקוראים יודעים לקרוא.

ולכן נצטרך לפצל את הקובץ ל chunks ולשלוח אותו במספר הודעות.

בכדי לשדר את המידע בין 2 המחשבים באופן שמבטיח שלא יאבד מידע, ושכל ההודעות יגיעו לפי   
הסדר הוטמע פרוטוקול הדומה לפרוטוקול מספר 2 שנלמד בקורס מבוא לתקשורת מחשבים.

התקשורת בין המחשבים היא סינכרונית,  
ובעצם הצמדנו לכל הודעה מה transmitter אינדקס שהוא מעגלי לפי mod10.  
כך כאשר ה receiver מקבל הודעה הוא יודע איזה מספר הודעה זו, והוא יודע שבהתחלה צריך לצפות להודעה מספר 1, ולאחר מכן 2 וכן הלאה... כל פעם הוא מציג ACK עבור ההודעה האחרונה שקיבל.

מכיוון שבקודי QR יש מנגנון checksum מובנה אין טעם להוסיף checksum משלנו על ההודעה, שכן אם הקורא הצליח לקרוא את ה QR סימן שהוא (לרוב המוחלט של הסיכויים) תקין.

6.1 – שיקולים בהצגת ה QR

יש כמה נקודות שרציתי להדגיש לגבי העברת הקובץ ב QR:

* ה transmitter מציג בתור QR Code ראשון את השם של הקובץ  
  זאת מכיוון שלכך מצפה ה receiver בכדי לדעת תחת איזה שם קובץ לשמור את הקובץ.
* בכדי לקרוא את הקובץ ולייצר QR Code אנו משתמשים בפונקציית גנרטור  
  זאת בכדי לא לקרוא את כל הקובץ לתוך הזכרון אף פעם.  
  גם ה receiver אינו טוען את כל תוכן הקובץ לזכרון אלא רושם אותו ב chunks.  
  זאת בכדי לשמור על Low Memory Usage.
* בפונקציית הגנרטור ה transmitter קורא 45 בתים כל פעם  
  מכיוון שאם נקרא יותר ייווצר QR Code גדול שה receiver לא תמיד מצליח לקרוא מתוך תמונת המצלמה.
* המידע ב QR Code מועבר ב base64  
  כלומר עובר base64 encoded string עם המידע הבינארי של הקובץ.
* התו הראשון בכל QR Code (גם בקורא וגם במשדר) מייצגים את האינדקס של ההודעה.  
  במקרה של המשדר זה אומר איזה הודעה הוא שידר כרגע, ובמקרה של הקורא זה לאיזה הודעה הוא עושה ACK. (כמובן שב ACK של הקורא אין יותר מידע חוץ מה INDEX).

6.2 – תרשים זרימה על אופן פעולת הקוד

תהליך השידור Transmitter:

transmitter.py

main()

Transmitter.transmit()

QR\_chunks\_generator

Read (at most) 45 bytes from the file

Wait for ACK

End of file

Display QR

When ACKed on the End Of Transmission QR - Exit

End of Transmission QR

Indexed Data QR

1

2

3a

3b

4

5

6

תהליך הקליטה Receiver:

Ack Filename QR

Read filename QR

Receiver

main()

Open in append mode file to receive

\_read\_qr\_contents\_to\_file()

End of transmission QR – Ack, wait 10 seconds and exit

ACK that QR’s index

Read QR from camera input

Increment ‘expected\_index’

7. תהליך שליחת קובץ

כאשר המשתמש רוצה לשלוח קובץ בין המחשבים, הוא מריץ את 2 הסקריפטים  
הסקריפט של transmitter פותח את הקובץ ומשדר סדרה של QR Codes על המסך.

רק אחרי שקיבל מה Receiver ברקוד שמסמל ACK על ההודעה, הוא עובר ל QR הבא.

כך עד שנגמר המידע ובו הוא משדר QR Code שמסמל End of transmission.

בכדי לראות הדגמה של איך זה נראה, תוכלו להסתכל תחת תיקיית tests/artifacts ולראות תמונות לדוגמה שהשתמשתי בהם בשביל הטסטים.

8. מקרי בדיקה

התכנית נבדקה לעומק ע״י כל מיני מקרים ידניים על כל סוגי הקבצים.  
ב repository מצורפים מספר מקרי בדיקה אוטומטיים שבהם אני מדמה את קלט המצלמה באמצעות תמונות מוכנות מראש (שהם לא צילומי מסך, אלא צילומים חיצוניים של מסך המחשב עצמו)

בכדי להריץ את הטסטים יש להריץ את הפקודה *pytest tests* מתוך תיקיית השורש של הפרוייקט.

9. פערים במצב הקיים

* העברת קובץ גדול היא איטית למדי שכן אנו קוראים ומשדרים 45 בתים מהקובץ בכל שידור.